

Who's yo DAHDI

An examination of HamVoIP theft

- Bryan Fields, W9CR
- 30-DEC-2019

Introductions

- app_rpt is a plug-in for Asterisk
- Zaptel does hardware interface
- Combine with radio = A
- Zapata became DAHDI
- Jim designed it all



app_rpt flow diagram Circa 2004

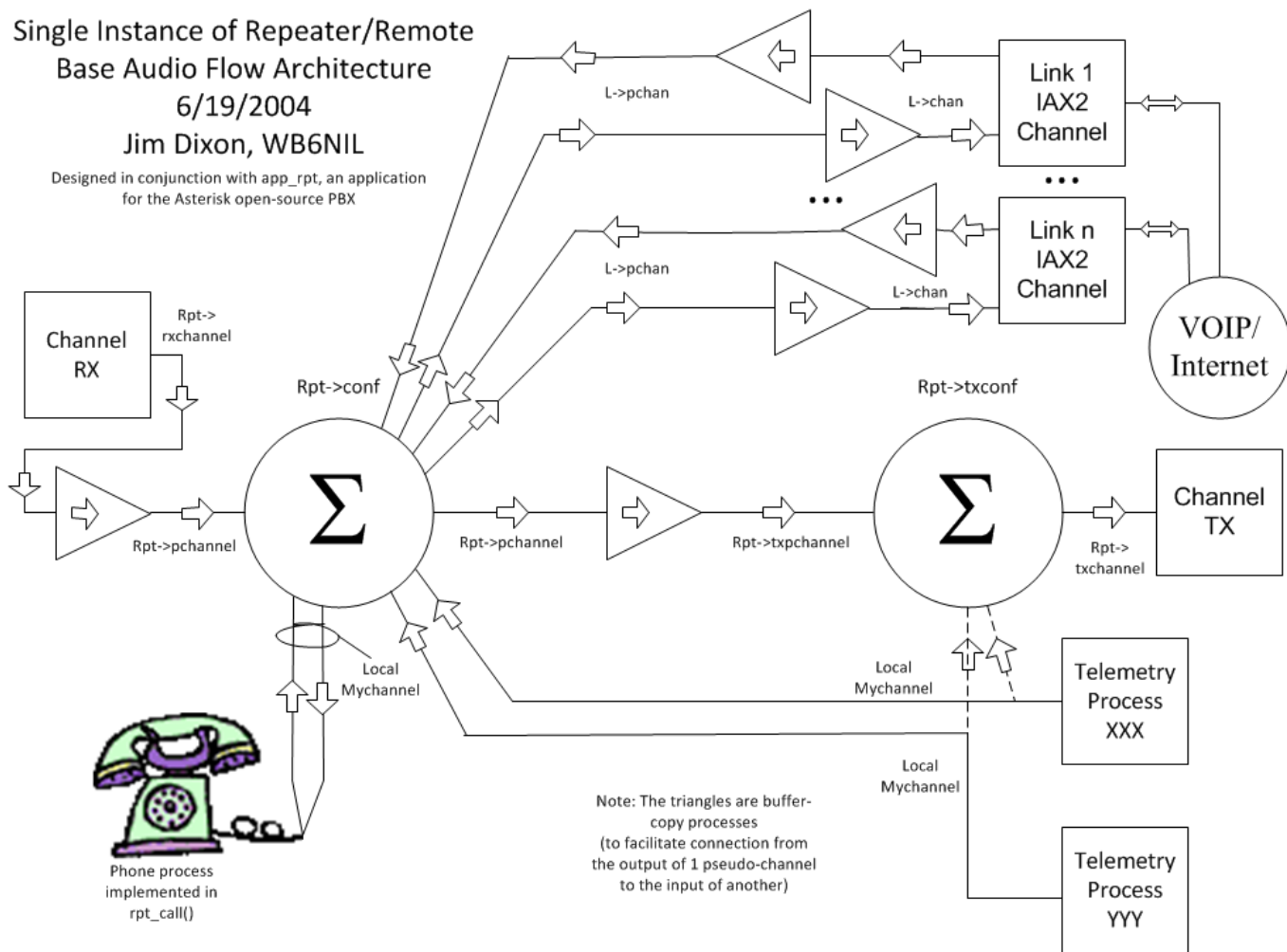
Single Instance of Repeater/Remote
Base Audio Flow Architecture
6/19/2004

Jim Dixon, WB6NIL

Designed in conjunction with app_rpt, an application
for the Asterisk open-source PBX

Timing Alignment
is important

Older hardware
included this as
the kernel lacked it
Newer kernels on
Intel don't have a
problem



Issues on ARM

- ARM - Acorn RISC Machine
- It's the Raspberry Pi!
- Asterisk (newer asterisk) runs well, Older no so much (1.4 is allstar)
- Why?

- ## Timing!

- Intel has this fixed, and this can cause stuttering and such in the conference calls that app_rpt uses
- Pickle Linux for the BeagleBoard-xM 2011 or so by Jim and N7PKT with the LoX interface
- The first ports of code to it RPi
 - 2013 - RPi 1 B+ - 700 MHz single core ARMv6, 512mb ram (runs like ass)
 - 2013 - Beagle Bone Black - 1GHz ARMv7, 512mb ram, works well as IO is not a factor
 - 2015 - RPi 2 - 900 MHz quad-core ARMv7, 1gb ram - first serious contender, USB issues
 - 2106 - RPi 3 - 1200 MHz quad-core ARMv7, 1gb ram - worked well

Porting to Arch Linux

Anthony, VK2ACP ports it and gets Zaptel/DAHDI running

[App_rpt-users] Arch Linux version of allstar has been stable

Anthony Percy [anthcp at gmail.com](mailto:anthcp@gmail.com)

Thu Jan 17 06:10:18 UTC 2013

- Previous message (by thread): [\[App_rpt-users\] Echolink](#)
- Next message (by thread): [\[App_rpt-users\] Arch Linux version of allstar has been stable for 2 weeks now....](#)
- Messages sorted by: [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

Hi all,
Just an update, the Arch Linux version of allstar has been stable for 2 weeks now on the repeater VK2RCZ.
If you are happy to compile the allstar code, I have updated my modified version to <https://github.com/anthcp/Arch-allstar>.
You may still have to fine tune the makefiles like add a symlink for the ar library but it does show the code changes....
We will make a Live CD in the future sometime....

Regards

Anthony, VK2ACP

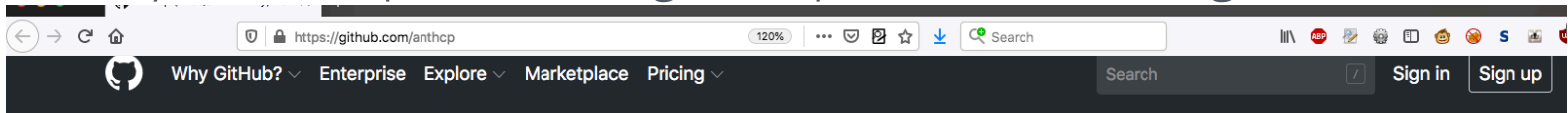
----- next part -----

An HTML attachment was scrubbed...

URL: <http://lists.keekles.org/pipermail/app_rpt-users/attachments/20130117/b7b89bea/attachment.html>

Porting to Arch Linux

Anthony, VK2ACP ports it and gets Zaptel/DAHDI running



Create your own GitHub profile Dismiss

Sign up for your own profile on GitHub, the best place to host code, manage projects, and build software alongside 40 million developers.

[Sign up](#)



VK5QS - Anthony
anthcp

Murray Bridge, Australia

[Block or report user](#)

Overview [Repositories 61](#) [Projects 0](#) [Stars 0](#) [Followers 2](#) [Following 4](#)

Popular repositories

[Dahdi-Arch-4.x-Linux-ARM-6and7-Allstar](#)

Dahdi Arch Linux 4.x device drivers for Allstar Asterisk

1 3

[LLVM-TMS320C64X](#)

Forked from alexjordan/LLVM-TMS320C64X

LLVM 2.9 branch with TI C64x backend.

[dahdi-linux-allstar-odroid](#)

[Allstar-dahdi-asterisk-arch-odroid](#)

[Allstar-using-dahdi-Arch-4x](#)

[nDPI](#)

Forked from nDPI/nDPI

Porting to ARM

First inquiries about Pi and BBB from Doug

[App_rpt-users] Asterisk and Allstar on the Raspberry Pi

Doug Crompton [doug at crompton.com](mailto:doug@crompton.com)

Sat Apr 20 17:35:54 UTC 2013

- Previous message (by thread): [\[App_rpt-users\] Voter Question](#)
- Next message (by thread): [\[App_rpt-users\] Asterisk and Allstar on the Raspberry Pi](#)
- Messages sorted by: [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

I have been using the Pi for awhile on IRLP with great success. It runs without a hitch. I was curious if anyone is using or has explored the use of the Pi with Asterisk on Allstar? It seems like the requirements would be similar. I use a USB sound FOB and the standard IRLP controller board. The Pi GPIO drives the IRLP controller in place of the standard PC parallel port.

I suspect the URI radio adapter or equivalent is all I would need in the way of hardware for Allstar. I have experience with Asterisk as I have been using it in a residential PBX for almost 10 years.

73 Doug
WA3DSP

<http://www.crompton.com/hamradio>

----- next part -----
An HTML attachment was scrubbed...
URL: <http://lists.keekles.org/pipermail/app_rpt-users/attachments/20130420/8269264a/attachment.html>

[App_rpt-users] Beagleboard Black and Allstar

Doug Crompton [doug at crompton.com](mailto:doug@crompton.com)

Thu Apr 25 23:21:34 UTC 2013

- Previous message (by thread): [\[App_rpt-users\] A completely off-topic request](#)
- Next message (by thread): [\[App_rpt-users\] Beagleboard Black and Allstar](#)
- Messages sorted by: [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

What are thoughts on using the Beaglebone Black in place of the Beagleboard for Allstar? It is about 1/3 of the price and it looks like it has as much or more capability.
<http://beagleboard.org/Products>

73 Doug
WA3DSP
<http://www.crompton.com/hamradio>

----- next part -----
An HTML attachment was scrubbed...
URL: <http://lists.keekles.org/pipermail/app_rpt-users/attachments/20130425/86f8176e/attachment.html>

David and Doug release a package for the BBB

June 2014

[App_rpt-users] BeagleBone Black Allstar Official Release

Doug Crompton doug_at_crompton.com

Thu Jun 26 05:02:00 UTC 2014

- Previous message (by thread): [\[App_rpt-users\] Question](#)
- Next message (by thread): [\[App_rpt-users\] BeagleBone Black Allstar Official Release](#)
- Messages sorted by: [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

Hello and welcome to the first public release of Allstar on the BeagleBone Black. This represents months of trial and error testing which first started with the Raspberry Pi back at the beginning of the year and was finally successful on the BBB platform. We have had several beta versions out now for over two months with many testers and we had very good reports. There is more to be done but this is a working system that will allow many to get rid of their energy consuming PC's with constantly running hard drives and have an Allstar system that fits in the palm of your hand.

The web page I have created gives very detailed instructions on how to download, install, and configure your BBB for use with Allstar. The web page is:

hamvoip.org

There you will find a link to download the image and also to join the arm-allstar email list. Please try to address questions specific to the BBB and Allstar to the arm-allstar list and not the apt-rpt list.

David and Doug release a package for the BBB

June 2014 – Jims follow up

[App_rpt-users] BeagleBone Black Allstar Official Release

Jim Duuuude [telesistant at hotmail.com](mailto:telesistant@hotmail.com)

Thu Jun 26 18:04:03 UTC 2014

- Previous message (by thread): [\[App_rpt-users\] BeagleBone Black Allstar Official Release](#)
- Next message (by thread): [\[App_rpt-users\] BeagleBone Black Allstar Official Release](#)
- Messages sorted by: [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)

It has quite sadly become painfully necessary to "Officially" and publicly state that this release is **NOT** an "Official" Allstar release. It has not been sanctioned by me or the Allstar network. It is something that this person is doing completely independently.

I am not in any way trying to infer that there is anything "right" or "wrong", or anything else about it, merely that is in an independent operation and I/we are not in any way responsible for its contents and/or use.

Jim WB6NIL

David and Doug release a package for the BBB

June 2014 – David ack's it is not AllStar

David McGough [kb4fxc at inttek.net](mailto:kb4fxc@inttek.net)

Thu Jun 26 22:03:33 UTC 2014

- Previous message (by thread): [\[App_rpt-users\] BeagleBone Black Allstar Official Release](#)
 - Next message (by thread): [\[App_rpt-users\] BeagleBone Black Allstar Official Release](#)
 - Messages sorted by: [\[date \]](#) [\[thread \]](#) [\[subject \]](#) [\[author \]](#)
-

Hi Jim and everyone,

I think Doug probably meant "officially released" since we've had several beta-test releases over the last 6 weeks or so.

I'm unfortunately not surprised by your comments, but, am disappointed. As the father of the AllStar project, you should be grinning from ear to ear that your baby has grown to the point that other developers are helping to build *your* network without requiring any of your time.

I, for one, hope this isn't another fork in the road. I hope it's just another step in the evolution of this already fantastic radio/repeater control and VoIP solution you envisioned.

As a software developer and project manager with decades of experience, one thing I've had to painfully learn is that everybody has there own vision for how a piece of software or a system should function. You can't stifle this creativity. Rather, you have to embrace it and direct it. If the system works reliably, inter-operates (with AllStar) compatibly and is being (pro)actively supported, then you smile and say: Good job guys! We need this feature next! Or, how about finding a solution for that issue???

This shouldn't be a competition. Lets work for a common goal.

Present Day September 2019

Skyler asks about hr_timer

On Wed, 24 Sep 2019, Skyler F wrote:

```
>
>
> Dahdi only runs well on a raspberry pi with Dahdi_Hrtimer.
> What does this do? How was this written? What is the reason why without
> this the raspberry pi has unreliable audio timing versus a normal
> computer?
>
> Other audio applications like audio / video streaming seem to run fine so
> why did you guys have to write something special for asterisk on the pi?
>
> Thanks
> Skyler
\
```

From: David McGough <kb4fxc@inttek.net>
Date: Wed, Sep 25, 2019 at 8:22 PM
Subject: Re: What does dahdi_hrtimer do?
To: Skyler F <electricity440@gmail.com>

Skyler,

I've spent literally hundreds of hours fixing timing issues in AllStar; Too much to even begin to try and describe in an e-mail. The dahdi_hrtimer code is one part of all the fixes. The Hrtimer driver was cobbled together by me.

This driver simulates (with lower resolution) the HPET timer functionality which X86 PC's have included in their design since about 2005. The ARM processor does not have any equivalent timer. So, I'm taking advantage of the ARM SDC counter found in the Broadcom SoC chips. The counter is free-running at 1MHz, with a 64-bit incrementing accumulator (so overflows are not a concern). I've adapted this timing source to the needs of AllStar / Asterisk---and, it works wonderfully.

As for why other audio/video applications work on the RPi?? Well, the demands of simply playing (or recording) an audio or video stream are simpler than what AllStar / Asterisk is doing. Asterisk is real-time mixing and perhaps transcoding audio from multiple sources. Each of these sources will have slightly different timing (meaning the stream bit rate will be a little fast or slow). An highly accurate clock is required to merge these streams, regenerating the frame timing in the process. Without this, you end up with pops, clicks, etc. What's happening underneath the covers is non-trivial.

What could this be?

It's part of DAHDI, no?

- First we boot it up and look around at the kernel modules

What could this be?

It's part of DAHDI, no?

- First we boot it up and look around at the kernel modules

```
[root@booty ~]# lsmod |grep dahdi
dahdi_hrtimer          16384  0
dahdi                  229376  69 dahdi_hrtimer
crc_ccitt              16384  1 dahdi
```

WOW, with it loaded!

```
[root@booty ~]# dahdi_test -c 20
Opened pseudo dahdi interface, measuring accuracy...
99.984% 99.989% 99.987% 99.988% 99.992% 99.991% 99.992% 99.994%
99.987% 99.989% 99.993% 99.988% 99.993% 99.988% 99.992% 99.990%
99.992% 99.993% 99.986% 99.991%
--- Results after 20 passes ---
Best: 99.994% -- Worst: 99.984% -- Average: 99.990017%
Cumulative Accuracy (not per pass): 99.990
```

What could this be?

It's part of DAHDI, no?

- Lets rmmod it and see

```
[root@booty ~]# rmmod dahdi_hrtimer
[root@booty ~]# dahdi_test -c 20
Opened pseudo dahdi interface, measuring accuracy...
99.458% 98.676% 98.519% 99.618% 99.621% 99.622% 98.519% 99.620%
99.618% 99.653% 99.600% 98.536% 99.620% 99.613% 99.449% 99.619%
99.620% 99.445% 99.619% 99.453%
--- Results after 20 passes ---
Best: 99.653% -- Worst: 98.519% -- Average: 99.374804%
Cummulative Accuracy (not per pass): 99.963
```

Yea, that's kind of normal for the RPi.

```
[root@booty ~]# cat /proc/sys/kernel/tainted
5120
```

Bits 10 and 12 are set, but the kernel is still all GPL:

<https://www.kernel.org/doc/html/latest/admin-guide/tainted-kernels.html>

Remember John David is much smarter than us

Where's the SOURCE!

- So lets decompress it and look at it
- `gunzip /usr/lib/modules/4.19.65-1-ARCH/dahdi/dahdi_hrtimer.ko.gz`
- `Readelf`
- Oh, it's got debugging symbols!
- It's GNU GPL code or so it says to the kernel!
- `.text` (the code) is only 360h or 864 bytes
- Lets strip it and see

```
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:                               ELF32
  Data:                               2's complement, little endian
  Version:                               1 (current)
  OS/ABI:                               UNIX - System V
  ABI Version:                           0
  Type:                                  REL (Relocatable file)
  Machine:                               ARM
  Version:                               0x1
  Entry point address:                   0x0
  Start of program headers:              0 (bytes into file)
  Start of section headers:              5524 (bytes into file)
  Flags:                                0x5000000, Version5 EABI
  Size of this header:                   52 (bytes)
  Size of program headers:               0 (bytes)
  Number of program headers:              0
  Size of section headers:               40 (bytes)
  Number of section headers:              24
  Section header string table index:      21

Section Headers:
[Nr] Name                               Type                               Addr      Off      Size    ES Flg Lk Inf Al
[ 0]                               NULL                               00000000  000000  000000  00      0  0  0
[ 1] .note.gnu.build-id                 NOTE                               00000000  000034  000024  00      A  0  0  4
[ 2] .text                             PROGBITS                          00000000  000058  000364  00     AX  0  0  4
[ 3] .rel.text                           REL                               00000000  00121c  000240  08     I 22  5  4
[ 4] .ARM.extab                          PROGBITS                          00000000  0003bc  000000  00      A  0  0  1
[ 5] .ARM.exidx                          ARM_EXIDX                          00000000  0003bc  000018  00     AL  2  0  4
[ 6] .rel.ARM.exidx                      REL                               00000000  00145c  000020  08     I 22  5  4
[ 7] .rodata                             PROGBITS                          00000000  0003d4  000076  00      A  0  0  4
[ 8] .rel.rodata                          REL                               00000000  00147c  000008  08     I 22  7  4
[ 9] .modinfo                             PROGBITS                          00000000  00044c  0000a4  00      A  0  0  4
[10] .rodata.str1.4                       PROGBITS                          00000000  0004f0  000249  01     AMS 0  0  4
[11] __param                             PROGBITS                          00000000  00073c  000014  00      A  0  0  4
[12] .rel__param                          REL                               00000000  001484  000020  08     I 22 11  4
[13] .note.Linux                          PROGBITS                          00000000  000750  000018  00      A  0  0  4
[14] .data                                PROGBITS                          00000000  000768  000000  00     WA  0  0  1
[15] .gnu.linkonce.thi                     PROGBITS                          00000000  000780  000200  00     WA  0  0 64
[16] .rel.gnu.linkonce                     REL                               00000000  0014a4  000010  08     I 22 15  4
[17] .bss                                 NOBITS                            00000000  000980  000040  00     WA  0  0  8
[18] .comment                             PROGBITS                          00000000  000980  000024  01     MS  0  0  1
[19] .note.GNU-stack                       PROGBITS                          00000000  0009a4  000000  00      0  0  1
[20] .ARM.attributes                       ARM_ATTRIBUTES                     00000000  0009a4  00002f  00      0  0  1
[21] .shstrtab                            STRTAB                            00000000  0014b4  0000e0  00      0  0  1
[22] .symtab                              SYMTAB                            00000000  0009d4  000560  10     23 63  4
[23] .strtab                              STRTAB                            00000000  00f34  0002e7  00      0  0  1
```

Remember John David is much smarter than me Where's the SOURCE!

- So lets decompress it and look at it
- `gunzip /usr/lib/modules/4.19.65-1-ARCH/dahdi/dahdi_hrtimer.ko.gz`
- `Readelf`
- Oh, it's got debugging symbols!
- It's GNU GPL code or so it says to the kernel!
- `.text` (the code) is only 360h or 864 bytes
- Lets strip it and see:

```
[root@booty ~]# strings dahdi_hrtimer.ko-stripped
dahdi_hrtimer_int
debug
license=GPL v2

description=Timing-Only Driver
parmtype=debug:int
depends=dahdi
name=dahdi_hrtimer
vermagic=4.19.65-1-ARCH SMP preempt mod_unload ARMv7 p2v8
5dahdi_hrtimer: HRTimer missed %lu ticks
7dahdi_hrtimer: 5000 ticks from hrtimer
3dahdi_hrtimer: Unable to allocate memory
dahdi_hrtimer
DAHDI_HRTIMER/1
%s (source: HRTimer) %d
DAHDI_HRTIMER/%d/%d
DAHDI Hi-Res Kernel Timing
3dahdi_hrtimer: Unable to initialize DAHDI driver (%d)
7dahdi_hrtimer: Trying to load High Resolution Timer
7dahdi_hrtimer: Initialized High Resolution Timer
7dahdi_hrtimer: Starting High Resolution Timer
6dahdi_hrtimer: High Resolution Timer started, good to go
7dahdi_hrtimer: init() finished
7dahdi_hrtimer: cleanup() finished
Linux
dahdi_hrtimer
GCC: (GNU) 5.3.0
GCC: (GNU) 5.3.0
aeabi
.shstrtab
.note.gnu.build-id
.text
.ARM.extab
.ARM.exidx
.rodata
.modinfo
.rodata.str1.4
__param
.note.Linux
.data
.gnu.linkonce.this_module
.bss
.comment
.note.GNU-stack
.ARM.attributes
```


Remember John David is much smarter than me Where's the SOURCE!

- So lets decompress it and look at it
- `gunzip /usr/lib/modules/4.19.65-1-ARCH/dahdi/dahdi_hrtimer.ko.gz`
- `Readelf`
- Oh, it's got debugging symbols!
- It's GNU GPL code or so it says to the kernel!
- `.text` (the code) is only 360h or 864 bytes
- Lets strip it and see:
- OhWhoa! Misspellings, I <3 misspellings
- It's initialize not **intialize**
- GOOGLE SEARCH!
 - "Unable to intialize DAHDI driver"

```
[root@booty ~]# strings dahdi_hrtimer.ko-stripped
dahdi_hrtimer_int
debug
license=GPL v2

description=Timing-Only Driver
parmtype=debug:int
depends=dahdi
name=dahdi_hrtimer
vermagic=4.19.65-1-ARCH SMP preempt mod_unload ARMv7 p2v8
5dahdi_hrtimer: HRTimer missed %lu ticks
7dahdi_hrtimer: 5000 ticks from hrtimer
3dahdi_hrtimer: Unable to allocate memory
dahdi_hrtimer
DAHDI_HRTIMER/1
%s (source: HRTimer) %d
DAHDI_HRTIMER/%d/%d
DAHDI Hi-Res Kernel Timing
3dahdi_hrtimer: Unable to intialize DAHDI driver (%d)
7dahdi_hrtimer: Trying to load High Resolution Timer
7dahdi_hrtimer: Initialized High Resolution Timer
7dahdi_hrtimer: Starting High Resolution Timer
6dahdi_hrtimer: High Resolution Timer started, good to go
7dahdi_hrtimer: init() finished
7dahdi_hrtimer: cleanup() finished
Linux
dahdi_hrtimer
GCC: (GNU) 5.3.0
GCC: (GNU) 5.3.0
aeabi
.shstrtab
.note.gnu.build-id
.text
.ARM.extab
.ARM.exidx
.rodata
.modinfo
.rodata.str1.4
__param
.note.Linux
.data
.gnu.linkonce.this_module
.bss
.comment
.note.GNU-stack
.ARM.attributes
```

Remember John David is much Where's the SOURCE!



"Unable to initialize DAHDI driver"



[All](#)

[Videos](#)

[Shopping](#)

[News](#)

[Images](#)

[More](#)

[Settings](#)

[Tools](#)

6 results (0.45 seconds)

[dahdi-linux/dahdi_dummy.c at master · asterisk/dahdi-linux ...](#)

[https://github.com](#) > [asterisk](#) > [dahdi-linux](#) > [blob](#) > [master](#) > [drivers](#) > [dah...](#) ▾
return -ENOMEM; } res = dahdi_dummy_initialize(ztd); if (res) { printk(KERN_ERR.
"dahdi_dummy: Unable to initialize DAHDI driver (%d)\n", res); kfree(ztd);

[Asterisk Guru](#)

[https://www.asteriskguru.com](#) > [archives](#) > [view-previous-topic-vt130850](#) ▾
Aug 27, 2008 - if (ztdummy_initialize(ztd)) { - printk(KERN_ERR "ztdummy: Unable to initialize
DAHDI driver\n"); + memset(ztd, 0x0, sizeof(struct dahdi_dummy));

[drivers/dahdi/dahdi_dummy.c - Debian Salsa](#)

[https://salsa.debian.org](#) > [pkg-voip-team](#) > [dahdi-firmware](#) > [blob](#) > [dahdi_...](#) ▾
... allocate memory\n"); return -ENOMEM; } if (dahdi_dummy_initialize(ztd)) { printk(KERN_ERR
"dahdi_dummy: Unable to initialize DAHDI driver\n"); kfree(ztd); ...

[File rtc.patch of Package dahdi-linux - openSUSE Build Service](#)

[https://build.opensuse.org](#) > [package](#) > [view_file](#) > [home:vitssoft:asterisk-11](#) ▾
... 0x0, sizeof(struct dahdi_dummy)); - if (dahdi_dummy_initialize(ztd)) { printk(KERN_ERR
"dahdi_dummy: Unable to initialize DAHDI driver\n"); kfree(ztd); return ...

[Loopback DAHDI Driver for DAHDI Telephony interface ...](#)

[https://moythreads.com](#) > [dahdi_loop](#) ▾
... if (dahdi_loop_initialize(dahdi_loop)) { printk("dahdi_loop: Unable to initialize dahdi
driver\n"); kfree(dahdi_loop); return -ENODEV; } printk(KERN_DEBUG ...

- So lets decompress it and look at it
- `gunzip /usr/lib/modules/4.19.65-1-ARCH/dahdi/dahdi_hrtimer.ko.gz`
- `Readelf`
- Oh, it's got debugging symbols!
- It's GNU GPL code or so it says to the kernel!
- `.text` (the code) is only 360h or 864 bytes
- Lets strip it and see:
- OhWhoa! Misspellings, I <3 misspellings
- It's initialize not **intialize**
- GOOGLE SEARCH!
 - "Unable to intialize DAHDI driver"

WTF, it's not just Dummy, is it?

dahdi_dummy is for timing only

- This hasn't been needed as dahdi has it built in, and Jim built the dummy driver in 2001
- Wait, only on intel. This be ARM.
- John David didn't just copy it and rename it
- Well lets look at the timer interface

```
[root@booty ~]# cat /proc/timer_list
Timer List Version: v0.8
HRTIMER_MAX_CLOCK_BASES: 8
now at 2224174991104 nsecs

cpu: 0
clock 0:
  .base:      85ed87e7
  .index:     0
  .resolution: 1 nsecs
  .get_time:   ktime_get
  .offset:     0 nsecs
active timers:
#0: HRTimer, dahdi_hrtimer_int, S:01
# expires at 2224175546145-2224175546145 nsecs [in 555041 to 555041 nsecs]
#1: <24a6cb73>, tick_sched_timer, S:01
# expires at 22241800000000-22241800000000 nsecs [in 5008896 to 5008896 nsecs]
#2: <4cade7ed>, hrtimer_wakeup, S:01
# expires at 2227634625901-2227639625899 nsecs [in 3459634797 to 3464634795 nsecs]
#3: <776f36fe>, timerfd_tmrproc, S:01
# expires at 2298933067000-2298933067000 nsecs [in 74758075896 to 74758075896 nsecs]
#4: sched_clock_timer, sched_clock_poll, S:01
# expires at 4398046511078-4398046511078 nsecs [in 2173871519974 to 2173871519974 nsecs]
clock 1:
  .base:      b1d3c4df
```

Can we compile dummy on ASL1.01?

Lets see if it works.

- Grabbed the code and compiled on ASL
 - Have to enable it in /usr/src/asl-dahdi-linux-2.11.1/linux/drivers/dahdi/Kbuild
- It built, so lets try it!
- Hell look at that:

```
root@StPeteRpt:/usr/src/asl-dahdi-linux-2.11.1/linux# dahdi_test -c 20
Opened pseudo dahdi interface, measuring accuracy...
99.985% 99.984% 99.991% 99.989% 99.990% 99.989% 99.989% 99.989%
99.990% 99.987% 99.985% 99.991% 99.991% 99.991% 99.991% 99.991%
99.991% 99.990% 99.991% 99.991%
--- Results after 20 passes ---
Best: 99.991% -- Worst: 99.984% -- Average: 99.989389%
Cumulative Accuracy (not per pass): 99.989
root@StPeteRpt:/usr/src/asl-dahdi-linux-2.11.1/linux#
```

```
root@StPeteRpt:/usr/src/asl-dahdi-linux-2.11.1/linux/drivers# ls -l
-rw-r--r-- 1 root root 8716 Oct  1 14:26 dahdi_dummy.ko
```

Can we compile dummy on ASL1.01?

Lets see if it works.

- Grabbed the code and compiled on ASL1.01
 - Have to enable it in /usr/src/asl-dahdi-lib
- It built, so lets try it!
- Hell look at that:
- This is not exact, the code is larger here in ASL1.01 (3e8 vs. 360 for hr)
- It is a different Kernel.
- Can we compile on HamVoIP and compare?

```
Start of program headers: 0 (bytes into file)
Start of section headers: 7676 (bytes into file)
Flags: 0x50000000, Version5 EABI
Size of this header: 52 (bytes)
Size of program headers: 0 (bytes)
Number of program headers: 0
Size of section headers: 40 (bytes)
Number of section headers: 26
Section header string table index: 25

Section Headers:
[Nr] Name                               Type           Addr           Off           Size          ES Flg Lk  Inf Al
[ 0]                               NULL           00000000       000000       000000       00  0  0  0  0
[ 1] .note.gnu.build-id                 NOTE           00000000       000034       000024       00  A  0  0  4
[ 2] .text                             PROGBITS       00000000       000058       0003e8       00  AX  0  0  4
[ 3] .rel.text                          REL            00000000       001a04       000280       08  I 23  2  4
[ 4] .ARM.extab                         PROGBITS       00000000       000440       000024       00  A  0  0  4
[ 5] .ARM.exidx                         ARM_EXIDX      00000000       000464       000018       00  AL  2  0  4
[ 6] .rel.ARM.exidx                     REL            00000000       001c84       000038       08  I 23  5  4
[ 7] .rodata                           PROGBITS       00000000       00047c       000076       00  A  0  0  4
[ 8] .rel.rodata                        REL            00000000       001cbc       000008       08  I 23  7  4
[ 9] .modinfo                           PROGBITS       00000000       0004f4       0000d9       00  A  0  0  4
[10] .rodata.str1.4                     PROGBITS       00000000       0005d0       00022f       01  AMS 0  0  4
[11] __param                            PROGBITS       00000000       000800       000014       00  A  0  0  4
[12] .rel__param                        REL            00000000       001cc4       000020       08  I 23 11  4
[13] __mcount_loc                       PROGBITS       00000000       000814       00000c       00  A  0  0  4
[14] .rel__mcount_loc                   REL            00000000       001ce4       000018       08  I 23 13  4
[15] __versions                         PROGBITS       00000000       000820       000600       00  A  0  0  4
[16] .data                              PROGBITS       00000000       000e20       000000       00  WA  0  0  1
[17] .gnu.linkonce.this_module          PROGBITS       00000000       000e40       000200       00  WA  0  0 64
[18] .rel.gnu.linkonce.this_module      REL            00000000       001cfc       000010       08  I 23 17  4
[19] .bss                               NOBITS         00000000       001040       000058       00  WA  0  0  8
[20] .comment                           PROGBITS       00000000       001040       00006a       01  MS  0  0  1
[21] .gnu.attributes                     PROGBITS       00000000       001040       000000       00  WA  0  0  0
```

Dahdi from ASL on HamVoIP

Nate's port is working

- Nate and Stacy got dahdi compiling on the newer kernel for the Pi 4
- I grabbed it and compiled it on HamVoIP
- Stripped of symbols, the files are the same size!
- Well that doesn't mean they are the same, lets look in IDA

```
[root@booty ~]# ls -al *stripped
-rw-r--r-- 1 root root 3384 Dec 29 22:08 dahdi_dummy.ko-stripped
-rw-r--r-- 1 root root 3384 Dec 29 22:06 dahdi_hrtimer.ko-stripped
```

```

root@booty ~]# readelf -e dahdi_hrtimer.ko-stripped
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:          ELF32
  Data:           2's complement, little endian
  Version:       1 (current)
  OS/ABI:        UNIX - System V
  ABI Version:   0
  Type:          REL (Relocatable file)
  Machine:       ARM
  Version:       0x1
  Entry point address: 0x0
  Start of program headers: 0 (bytes into file)
  Start of section headers: 2704 (bytes into file)
  Flags:         0x5000000, Version5 EABI
  Size of this header: 52 (bytes)
  Size of program headers: 0 (bytes)
  Number of program headers: 0
  Size of section headers: 40 (bytes)
  Number of section headers: 17
  Section header string table index: 16

```

Section Headers:

[Nr]	Name	Type	Addr	Off	Size	ES	Flg	L
[0]		NULL	00000000	000000	000000	00		
[1]	.note.gnu.build-id	NOTE	00000000	000034	000024	00	A	
[2]	.text	PROGBITS	00000000	000058	000364	00	AX	
[3]	.ARM.extab	PROGBITS	00000000	0003bc	000000	00	A	
[4]	.ARM.exidx	ARM_EXIDX	00000000	0003bc	000018	00	AL	
[5]	.rodata	PROGBITS	00000000	0003d4	000076	00	A	
[6]	.modinfo	PROGBITS	00000000	00044c	0000a4	00	A	
[7]	.rodata.str1.4	PROGBITS	00000000	0004f0	000249	01	AMS	
[8]	__param	PROGBITS	00000000	00073c	000014	00	A	
[9]	.note.Linux	PROGBITS	00000000	000750	000018	00	A	
[10]	.data	PROGBITS	00000000	000768	000000	00	WA	
[11]	.gnu.linkonce.thi	PROGBITS	00000000	000780	000200	00	WA	
[12]	.bss	NOBITS	00000000	000980	000040	00	WA	
[13]	.comment	PROGBITS	00000000	000980	000024	01	MS	
[14]	.note.GNU-stack	PROGBITS	00000000	0009a4	000000	00		
[15]	.ARM.attributes	ARM_ATTRIBUTES	00000000	0009a4	00002f	00		
[16]	.shstrtab	STRTAB	00000000	0009d3	0000bc	00		

Key to Flags:

W (write), A (alloc), X (execute), M (merge), S (strings)
 I (info), L (link order), G (group), T (TLS), E (exclude), x (unknown)
 0 (extra OS processing required) o (OS specific), p (processor specific)

```

[[root@booty ~]# readelf -e dahdi_dummy.ko-stripped
ELF Header:
  Magic:   7f 45 4c 46 01 01 01 00 00 00 00 00 00 00 00 00
  Class:          ELF32
  Data:           2's complement, little endian
  Version:       1 (current)
  OS/ABI:        UNIX - System V
  ABI Version:   0
  Type:          REL (Relocatable file)
  Machine:       ARM
  Version:       0x1
  Entry point address: 0x0
  Start of program headers: 0 (bytes into file)
  Start of section headers: 2704 (bytes into file)
  Flags:         0x5000000, Version5 EABI
  Size of this header: 52 (bytes)
  Size of program headers: 0 (bytes)
  Number of program headers: 0
  Size of section headers: 40 (bytes)
  Number of section headers: 17
  Section header string table index: 16

```

Section Headers:

[Nr]	Name	Type	Addr	Off	Size	ES	Flg	Lk	Inf	Al
[0]		NULL	00000000	000000	000000	00		0	0	0
[1]	.note.gnu.build-id	NOTE	00000000	000034	000024	00	A	0	0	4
[2]	.text	PROGBITS	00000000	000058	000364	00	AX	0	0	4
[3]	.ARM.extab	PROGBITS	00000000	0003bc	000000	00	A	0	0	1
[4]	.ARM.exidx	ARM_EXIDX	00000000	0003bc	000018	00	AL	2	0	4
[5]	.rodata	PROGBITS	00000000	0003d4	000076	00	A	0	0	4
[6]	.modinfo	PROGBITS	00000000	00044c	0000c6	00	A	0	0	4
[7]	.rodata.str1.4	PROGBITS	00000000	000514	00022f	01	AMS	0	0	4
[8]	__param	PROGBITS	00000000	000744	000014	00	A	0	0	4
[9]	.note.Linux	PROGBITS	00000000	000758	000018	00	A	0	0	4
[10]	.data	PROGBITS	00000000	000770	000000	00	WA	0	0	1
[11]	.gnu.linkonce.thi	PROGBITS	00000000	000780	000200	00	WA	0	0	64
[12]	.bss	NOBITS	00000000	000980	000040	00	WA	0	0	8
[13]	.comment	PROGBITS	00000000	000980	000024	01	MS	0	0	1
[14]	.note.GNU-stack	PROGBITS	00000000	0009a4	000000	00		0	0	1
[15]	.ARM.attributes	ARM_ATTRIBUTES	00000000	0009a4	00002f	00		0	0	1
[16]	.shstrtab	STRTAB	00000000	0009d3	0000bc	00		0	0	1

Key to Flags:

W (write), A (alloc), X (execute), M (merge), S (strings)
 I (info), L (link order), G (group), T (TLS), E (exclude), x (unknown)
 0 (extra OS processing required) o (OS specific), p (processor specific)

Functions window

- dahdi_hrtimer_int
- init_module
- cleanup_module
- dev_set_name
- snprintf
- hrtimer_init
- _printk_ratelimit
- kfree
- kmem_cache_alloc_trace
- _dahdi_receive
- dahdi_create_device
- printk
- dahdi_unregister_device
- dahdi_register_device
- hrtimer_start_range_ns
- sprintf
- hrtimer_cancel
- hrtimer_forward
- dahdi_free_device
- _dahdi_transmit

Line 1 of 20

Graph overview

IDA View-A

Hex View-1

Structures

Enums

Functions window

Function name

- dahdi_dummy_hr_int
- init_module
- cleanup_module
- dev_set_name
- snprintf
- hrtimer_init
- _printk_ratelimit
- kfree
- kmem_cache_alloc_trace
- _dahdi_receive
- dahdi_create_device
- printk
- dahdi_unregister_device
- dahdi_register_device
- hrtimer_start_range_ns
- sprintf
- hrtimer_cancel
- hrtimer_forward
- dahdi_free_device
- _dahdi_transmit

Line 15 of 20

Graph overview

Output window

```

; Segment type: Pure code
AREA .text, CODE
CODE32

dahdi_hrtimer_int

var_20= -0x20

STMFD      SP!, {R4-R7,LR}
MOV        R4, #ztd
MOV        R6, R0
LDR        R0, [R4]
SUB        SP, SP, #0xC
ADD        R0, R0, #4
MRS        R5, CPSR
CPSID      I
BL         _dahdi_receive
BIC        R5, R5, #0x40
MRS        R3, CPSR
AND        R3, R3, #0x40
ORR        R5, R3, R5
MSR        CPSR_c, R5
LDR        R0, [R4]
ADD        R0, R0, #4
MRS        R5, CPSR
CPSID      I
BL         _dahdi_transmit
BIC        R5, R5, #0x40
MRS        R3, CPSR
AND        R3, R3, #0x40
ORR        R5, R3, R5
MSR        CPSR_c, R5
LDRD       R2, [R6,#0x10]
MOV        R6, #0xF4240
MOV        R7, #0
ADD        R0, R4, #8
STRD       R6, [SP,#0x20+var_20]
BL         hrtimer_forward
CMP        R0, #1
BLS        loc_A0

MOV        R5, R0
MOV        R0, #_func_.33590 ; "dahdi_hrtimer_int"
BL         _printk_ratelimit
CMP        R0, #0

```

100.00% | (-40,360) | (241,6) | 00000008 | 000000000000000028: dahdi_hrtimer_int

IDA View-A

Hex View-1

Structures

Enums

Imports

Functions window

Function name

- dahdi_dummy_hr_int
- init_module
- cleanup_module
- dev_set_name
- snprintf
- hrtimer_init
- _printk_ratelimit
- kfree
- kmem_cache_alloc_trace
- _dahdi_receive
- dahdi_create_device
- printk
- dahdi_unregister_device
- dahdi_register_device
- hrtimer_start_range_ns
- sprintf
- hrtimer_cancel
- hrtimer_forward
- dahdi_free_device
- _dahdi_transmit

Line 15 of 20

Graph overview

Output window

```

; Segment type: Pure code
AREA .text, CODE
CODE32

dahdi_dummy_hr_int

var_20= -0x20

STMFD      SP!, {R4-R7,LR}
MOV        R4, #ztd
MOV        R6, R0
LDR        R0, [R4]
SUB        SP, SP, #0xC
ADD        R0, R0, #4
MRS        R5, CPSR
CPSID      I
BL         _dahdi_receive
BIC        R5, R5, #0x40
MRS        R3, CPSR
AND        R3, R3, #0x40
ORR        R5, R3, R5
MSR        CPSR_c, R5
LDRD       R2, [R6,#0x10]
MOV        R6, #0xF4240
MOV        R7, #0
ADD        R0, R4, #8
STRD       R6, [SP,#0x20+var_20]
BL         hrtimer_forward
CMP        R0, #1
BLS        loc_A0

MOV        R5, R0
MOV        R0, #_func_.33585 ; "dahdi_dummy_hr_int"
BL         _printk_ratelimit
CMP        R0, #0

```

100.00% | (-22,349) | (4,585) | 00000168 | 0000000000000000110: dahdi_dummy_hr_int+110 (Synchro

IDA View-A

Hex View-1

Structures

Enums

Imports

Functions window

Function name

- dahdi_dummy_hr_int
- init_module
- cleanup_module
- dev_set_name
- snprintf
- hrtimer_init
- _printk_ratelimit
- kfree
- kmem_cache_alloc_trace
- _dahdi_receive
- dahdi_create_device
- printk
- dahdi_unregister_device
- dahdi_register_device
- hrtimer_start_range_ns
- sprintf
- hrtimer_cancel
- hrtimer_forward
- dahdi_free_device
- _dahdi_transmit

Line 15 of 20

Graph overview

Output window

```

; Segment type: Pure code
AREA .text, CODE
CODE32

dahdi_dummy_hr_int

var_20= -0x20

STMFD      SP!, {R4-R7,LR}
MOV        R4, #ztd
MOV        R6, R0
LDR        R0, [R4]
SUB        SP, SP, #0xC
ADD        R0, R0, #4
MRS        R5, CPSR
CPSID      I
BL         _dahdi_receive
BIC        R5, R5, #0x40
MRS        R3, CPSR
AND        R3, R3, #0x40
ORR        R5, R3, R5
MSR        CPSR_c, R5
LDRD       R2, [R6,#0x10]
MOV        R6, #0xF4240
MOV        R7, #0
ADD        R0, R4, #8
STRD       R6, [SP,#0x20+var_20]
BL         hrtimer_forward
CMP        R0, #1
BLS        loc_A0

MOV        R5, R0
MOV        R0, #_func_.33585 ; "dahdi_dummy_hr_int"
BL         _printk_ratelimit
CMP        R0, #0

```

100.00% | (-22,349) | (4,585) | 00000168 | 0000000000000000110: dahdi_dummy_hr_int+110 (Synchro

Dahdi from ASL on HamVoIP

Nate's port is working

- Holy shit, it's the same damn code.
- The only differences are it was renamed and authors removed
- John David's M.O.
- Lol, I spent more time on this video
- Lets look at the assembled code
Just to be sure there's nothing hidden in
his kernel module
- `objdump -d dahdi_hrtimer.ko-stripped`

I've spent literally hundreds of hours fixing timing issues in AllStar;
Too much to even begin to try and describe in an e-mail. The dahdi_hrtimer
code is one part of all the fixes. The Hrtimer driver was cobbled together
by me.

dahdi_hrtimer.ko-stripped: file format elf32-littlearm

Disassembly of section .text:

00000000 <.text>:

```
0: e92d40f0 push {r4, r5, r6, r7, lr}
4: e3004000 movw r4, #0
8: e3404000 movt r4, #0
c: e1a06000 mov r6, r0
10: e5940000 ldr r0, [r4]
14: e24dd00c sub sp, sp, #12
18: e2800004 add r0, r0, #4
1c: e10f5000 mrs r5, CPSR
20: f10c0080 cpsid i
24: ebfffffe bl 0x24
28: e3c55040 bic r5, r5, #64 ; 0x40
2c: e10f3000 mrs r3, CPSR
30: e2033040 and r3, r3, #64 ; 0x40
34: e1835005 orr r5, r3, r5
38: e121f005 msr CPSR_c, r5
3c: e5940000 ldr r0, [r4]
40: e2800004 add r0, r0, #4
44: e10f5000 mrs r5, CPSR
48: f10c0080 cpsid i
4c: ebfffffe bl 0x4c
50: e3c55040 bic r5, r5, #64 ; 0x40
54: e10f3000 mrs r3, CPSR
58: e2033040 and r3, r3, #64 ; 0x40
5c: e1835005 orr r5, r3, r5
60: e121f005 msr CPSR_c, r5
64: e1c621d0 ldrd r2, [r6, #16]
68: e3046240 movw r6, #16960 ; 0x4240
6c: e340600f movt r6, #15
```

dahdi_dummy.ko-stripped: file format elf32-littlearm

Disassembly of section .text:

00000000 <.text>:

```
0: e92d40f0 push {r4, r5, r6, r7, lr}
4: e3004000 movw r4, #0
8: e3404000 movt r4, #0
c: e1a06000 mov r6, r0
10: e5940000 ldr r0, [r4]
14: e24dd00c sub sp, sp, #12
18: e2800004 add r0, r0, #4
1c: e10f5000 mrs r5, CPSR
20: f10c0080 cpsid i
24: ebfffffe bl 0x24
28: e3c55040 bic r5, r5, #64 ; 0x40
2c: e10f3000 mrs r3, CPSR
30: e2033040 and r3, r3, #64 ; 0x40
34: e1835005 orr r5, r3, r5
38: e121f005 msr CPSR_c, r5
3c: e5940000 ldr r0, [r4]
40: e2800004 add r0, r0, #4
44: e10f5000 mrs r5, CPSR
48: f10c0080 cpsid i
4c: ebfffffe bl 0x4c
50: e3c55040 bic r5, r5, #64 ; 0x40
54: e10f3000 mrs r3, CPSR
58: e2033040 and r3, r3, #64 ; 0x40
5c: e1835005 orr r5, r3, r5
60: e121f005 msr CPSR_c, r5
64: e1c621d0 ldrd r2, [r6, #16]
68: e3046240 movw r6, #16960 ; 0x4240
6c: e340600f movt r6, #15
```

68:	e3046240	movw	r6, #16960	; 0x4240	68:	e3046240	movw	r6, #16960	; 0x4240
6c:	e340600f	movt	r6, #15		6c:	e340600f	movt	r6, #15	
70:	e3a07000	mov	r7, #0		70:	e3a07000	mov	r7, #0	
74:	e2840008	add	r0, r4, #8		74:	e2840008	add	r0, r4, #8	
78:	e1cd60f0	strd	r6, [sp]		78:	e1cd60f0	strd	r6, [sp]	
7c:	ebffffffe	bl	0x7c		7c:	ebffffffe	bl	0x7c	
80:	e3500001	cmp	r0, #1		80:	e3500001	cmp	r0, #1	
84:	9a000005	bls	0xa0		84:	9a000005	bls	0xa0	
88:	e1a05000	mov	r5, r0		88:	e1a05000	mov	r5, r0	
8c:	e3000000	movw	r0, #0		8c:	e3000000	movw	r0, #0	
90:	e3400000	movt	r0, #0		90:	e3400000	movt	r0, #0	
94:	ebffffffe	bl	0x94		94:	ebffffffe	bl	0x94	
98:	e3500000	cmp	r0, #0		98:	e3500000	cmp	r0, #0	
9c:	1a000017	bne	0x100		9c:	1a000013	bne	0xf0	
a0:	e5943038	ldr	r3, [r4, #56]	; 0x38	a0:	e5942038	ldr	r2, [r4, #56]	; 0x38
a4:	e3002000	movw	r2, #0		a4:	e3003000	movw	r3, #0	
a8:	e3402000	movt	r2, #0		a8:	e3403000	movt	r3, #0	
ac:	e3530001	cmp	r3, #1		ac:	e3520000	cmp	r2, #0	
b0:	da00000b	ble	0xe4		b0:	0a00000b	beq	0xe4	
b4:	e592303c	ldr	r3, [r2, #60]	; 0x3c	b4:	e593103c	ldr	r1, [r3, #60]	; 0x3c
b8:	e3080bad	movw	r0, #35757	; 0x8bad	b8:	e3082bad	movw	r2, #35757	; 0x8bad
bc:	e34608db	movt	r0, #26843	; 0x68db	bc:	e34628db	movt	r2, #26843	; 0x68db
c0:	e301c388	movw	ip, #5000	; 0x1388	c0:	e3010388	movw	r0, #5000	; 0x1388
c4:	e2831001	add	r1, r3, #1		c4:	e281c001	add	ip, r1, #1	
c8:	e582103c	str	r1, [r2, #60]	; 0x3c	c8:	e583c03c	str	ip, [r3, #60]	; 0x3c
cc:	e0c10093	smull	r0, r1, r3, r0		cc:	e0c32291	smull	r2, r3, r1, r2	
d0:	e1a02fc3	asr	r2, r3, #31		d0:	e1a02fc1	asr	r2, r1, #31	
d4:	e06225c1	rsb	r2, r2, r1, asr #11		d4:	e06225c3	rsb	r2, r2, r3, asr #11	
d8:	e063329c	mls	r3, ip, r2, r3		d8:	e0631290	mls	r3, r0, r2, r1	
dc:	e3530000	cmp	r3, #0		dc:	e3530000	cmp	r3, #0	
e0:	0a000002	beq	0xf0		e0:	0a000007	beq	0x104	
e4:	e3a00001	mov	r0, #1		e4:	e3a00001	mov	r0, #1	
e8:	e28dd00c	add	sp, sp, #12		e8:	e28dd00c	add	sp, sp, #12	
ec:	e8bd80f0	pop	{r4, r5, r6, r7, pc}		ec:	e8bd80f0	pop	{r4, r5, r6, r7, pc}	
f0:	e3000000	movw	r0, #0		f0:	e3000000	movw	r0, #0	

2e0:	ebffffffe	bl	0x2e0
2e4:	e3000000	movw	r0, #0
2e8:	e3400000	movt	r0, #0
2ec:	ebffffffe	bl	0x2ec
2f0:	e5954038	ldr	r4, [r5, #56] ; 0x38
2f4:	e3540000	cmp	r4, #0
2f8:	0afffffd6	beq	0x258
2fc:	e3000000	movw	r0, #0
300:	e1a04007	mov	r4, r7
304:	e3400000	movt	r0, #0
308:	ebffffffe	bl	0x308
30c:	eaaffffd1	b	0x258
310:	00000014	andeq	r0, r0, r4, lsl r0
314:	e92d4010	push	{r4, lr}
318:	e3004000	movw	r4, #0
31c:	e3404000	movt	r4, #0
320:	e2840008	add	r0, r4, #8
324:	ebffffffe	bl	0x324
328:	e5943000	ldr	r3, [r4]
32c:	e5930000	ldr	r0, [r3]
330:	ebffffffe	bl	0x330
334:	e5943000	ldr	r3, [r4]
338:	e5930000	ldr	r0, [r3]
33c:	ebffffffe	bl	0x33c
340:	e5940000	ldr	r0, [r4]
344:	ebffffffe	bl	0x344
348:	e5943038	ldr	r3, [r4, #56] ; 0x38
34c:	e3530000	cmp	r3, #0
350:	08bd8010	popeq	{r4, pc}
354:	e3000000	movw	r0, #0
358:	e3400000	movt	r0, #0
35c:	e8bd4010	pop	{r4, lr}
360:	eaafffffe	b	0x360

2e0:	ebffffffe	bl	0x2e0
2e4:	e3000000	movw	r0, #0
2e8:	e3400000	movt	r0, #0
2ec:	ebffffffe	bl	0x2ec
2f0:	e5954038	ldr	r4, [r5, #56] ; 0x38
2f4:	e3540000	cmp	r4, #0
2f8:	0afffffd6	beq	0x258
2fc:	e3000000	movw	r0, #0
300:	e1a04007	mov	r4, r7
304:	e3400000	movt	r0, #0
308:	ebffffffe	bl	0x308
30c:	eaaffffd1	b	0x258
310:	00000014	andeq	r0, r0, r4, lsl r0
314:	e92d4010	push	{r4, lr}
318:	e3004000	movw	r4, #0
31c:	e3404000	movt	r4, #0
320:	e2840008	add	r0, r4, #8
324:	ebffffffe	bl	0x324
328:	e5943000	ldr	r3, [r4]
32c:	e5930000	ldr	r0, [r3]
330:	ebffffffe	bl	0x330
334:	e5943000	ldr	r3, [r4]
338:	e5930000	ldr	r0, [r3]
33c:	ebffffffe	bl	0x33c
340:	e5940000	ldr	r0, [r4]
344:	ebffffffe	bl	0x344
348:	e5943038	ldr	r3, [r4, #56] ; 0x38
34c:	e3530000	cmp	r3, #0
350:	08bd8010	popeq	{r4, pc}
354:	e3000000	movw	r0, #0
358:	e3400000	movt	r0, #0
35c:	e8bd4010	pop	{r4, lr}
360:	eaafffffe	b	0x360

Dahdi from ASL on HamVoIP

Does it do the same thing?

- It's 100% the same code at the ASM level
- But does it do the same thing?

```
300:  eartime 0 0x300
[[root@booty ~]# lsmod
Module                Size  Used by
dahdi_hrtimer         16384  0
bcmfmac               225280  0
bcmutil               16384  1 bcmfmac
```

Dahdi from ASL on HamVoIP

Does it do the same thing?

- It's 100% the same code at the ASM level
- But does it do the same thing?
- Yes

```
Asterisk 1.4.23-pre.hamvoip-V1.5.4-03-app_rpt-0.327-10/22/2019, Copyright (C) 1999 - 2019 HamVoIP.org and others.  
Created by Mark Spencer <markster@digium.com>  
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.  
This is free software, with components licensed under the GNU General Public  
License version 2 and other licenses; you are welcome to redistribute it under  
certain conditions. Type 'core show license' for details.  
=====
```

Connected to Asterisk 1.4.23-pre.hamvoip-V1.5.4-03-app_rpt-0.327-10/22/2019 currently running on booty (pid = 323)
Verbosity is at least 3
booty*CLI> dahdi show
cadences channel channels status
booty*CLI> dahdi show status

Description	Alarms	IRQ	bpviol	CRC4
DAHDI_HRTIMER/1 (source: HRTimer) 1	UNCONFIGUR	0	0	0

Dahdi from ASL on HamVoIP

Does it do the same thing?

- It's 100% the same code at the ASM level
- But does it do the same thing?
- Yes

```
[root@booty ~]# rmmod dahdi_hrtimer
[root@booty ~]# asterisk -r
Asterisk 1.4.23-pre.hamvoip-V1.5.4-03-app_rpt-0.327-10/22/2019, Copyright (C) 1999 - 2019 HamVoIP.org and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.4.23-pre.hamvoip-V1.5.4-03-app_rpt-0.327-10/22/2019 currently running on booty (pid = 323)
Verbosity is at least 3
booty*CLI> dahdi show status
Description                               Alarms    IRQ       bpviol    CRC4
booty*CLI> quit
```

Removed
dahdi_hrtimer.ko

Dahdi from ASL on HamVoIP

Does it do the same thing?

- It's 100% the same code at the ASM level
- But does it do the same thing?
- Yes

```
[root@booty ~]# asterisk -r
Asterisk 1.4.23-pre.hamvoip-V1.5.4-03-app_rpt-0.327-10/22/2019, Copyright (C) 1999 - 2019 HamVoIP.org and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 1.4.23-pre.hamvoip-V1.5.4-03-app_rpt-0.327-10/22/2019 currently running on booty (pid = 323)
Verbosity is at least 3
booty*CLI> dahdi show status
Description                               Alarms      IRQ      bpviol      CRC4
DAHDI_DUMMY/1 (source: HRtimer) 1        UNCONFIGUR  0           0           0
booty*CLI> quit
[root@booty ~]# dahdi_test -c 20
Opened pseudo dahdi interface, measuring accuracy...
99.983% 99.981% 99.992% 99.992% 99.990% 99.991% 99.993% 99.992%
99.993% 99.986% 99.990% 99.993% 99.992% 99.993% 99.989% 99.990%
99.993% 99.991% 99.992% 99.990%
--- Results after 20 passes ---
Best: 99.993% -- Worst: 99.981% -- Average: 99.990274%
Cumulative Accuracy (not per pass): 99.990
```

Installed
dahdi_dummy.ko

Summary

- John David McGough, KB4FXC is a thief.
- There is no way to spin this other than he is passing off other
- This isn't an isolated incident.
- Internet Technologies, Inc. c
- I reached out to the listed author, and he had no idea about



Summary

- John David McGough, KB4FXC is a thief.
- There is no way to spin this other than he is passing off others work as his own
- This isn't an isolated incident.
 - Internet Technologies, Inc. customers beware!
- I reached out to the listed author, and he had no idea about this, no permission was granted for it.
- **Asterisk on Arch was created by others**
- Wait! How do I enable this on my ASL1.01 RPi node?
 - <https://wiki.w9cr.net>

Questions?